Ліхачова В. КН-19-1

**Лабораторна робота №4**

**Рефакторинг програмного коду**

**Мета роботи**: навчитися виконувати реорганізацію програмного коду на підставі шаблонів рефакторінгу.

**Завдання**

Виконати аналіз програмного коду розробленого ПО і модульних тестів з метою виявлення погано організованого коду. Використовуючи шаблони рефакторінгу, виконати реорганізацію програмного коду розробленого ПО і модульних тестів. Виконати опис вироблених операцій рефакторінгу.

**Хід роботи**:

***Додавання параметра (Add Parameter)***

***Метод рефакторингу: Виділення батьківського класу (Extract Superclass)***

***Проблема***

*У вас є два класи із загальними полями та методами.*

***Рішення***

*Створіть для них загальний суперклас та перенесіть туди однакові поля та методи.*

**БУЛО:**

class Provider

{

public Provider(string name, string address, uint supply, double cost) //Поставщик

{

this.name = name;

this.address = address;

this.supply = supply;

this.cost = cost;

}

String name;

String address;

public uint supply; //Запасы

public double cost; //Стоимость товара

}

class Client //Покупатель

{

public Client(string name,string address, uint order)

{

this.name = name;

this.address = address;

this.order = order;

}

string name;

string address;

public uint order; //потреби споживача

}

**СТАЛО:**

abstract public class general\_information

{

public string name;

public string address;

public string removing\_extra\_spaces(string str)

{

var newstr = string.Join(" ", str.Split(new[] { ' ' }, StringSplitOptions.RemoveEmptyEntries).Select(item => item.Trim()));

return newstr;

}

public string fix\_the\_registry(string str)

{

string[] newstr = str.Split();

for (int i = 0; i < newstr.Length; i++)

{

newstr[i] = char.ToUpper(newstr[i][0]) + newstr[i].Substring(1).ToLower();

}

str = string.Join(" ", newstr);

return str;

}

public string return\_address()

{

return address;

}

}

public class Provider : general\_information

{

public Provider(String name\_provider, String address\_provider, uint supply, double cost) //Постачальник

{

name\_provider = removing\_extra\_spaces(name\_provider);

address\_provider = removing\_extra\_spaces(address\_provider);

name\_provider = fix\_the\_registry(name\_provider);

address\_provider = fix\_the\_registry(address\_provider);

this.name = name\_provider;

this.address = address\_provider;

this.supply = supply;

this.cost = cost;

}

public uint supply; //Запаси

public double cost; //Вартість товару

~Provider()

{

}

}

public class Client : general\_information //Клієнт

{

public Client(string name\_client, string address\_client, uint order)

{

name\_client = removing\_extra\_spaces(name\_client);

address\_client = removing\_extra\_spaces(address\_client);

name\_client = fix\_the\_registry(name\_client);

address\_client = fix\_the\_registry(address\_client);

this.name = name\_client;

this.order = order;

this.address = address\_client;

}

public uint order; //потреби споживача

~Client()

{

}

}

***Метод рефакторингу: Приховування методу (Hide Method)***

***Проблема***

*Метод не використовується іншими класами або використовується лише усередині своєї ієрархії класів.*

***Рішення***

*Зробіть метод приватним чи захищеним.*

**БУЛО:**

public class OptimalPlan //опорний план

{

***…***

public OptimalPlan(double[,] \_cost, uint[] \_stock, uint[] \_needs)

{

cost = \_cost;

stock = \_stock;

order = \_needs;

Initialization();

}

private void Initialization()…

public void Start\_calculating()

{

Metod\_MinimalElement();

FindingPotentials();

}

***…***

}

**СТАЛО**

public class OptimalPlan //опорний план

{

**…**

public OptimalPlan(double[,] \_cost, uint[] \_stock, uint[] \_needs)

{

cost = \_cost;

stock = \_stock;

order = \_needs;

Initialization();

Start\_calculating();

}

private void Initialization()…

private void Start\_calculating()

{

Metod\_MinimalElement();

FindingPotentials();

}

**…**

}

***Метод рефакторингу****:* ***Extract Method***

***Проблема***

*У вас є фрагмент коду, який можна згрупувати.*

***Рішення***

*Виділіть ділянку коду в новий метод (або функцію) та викличте цей метод замість старого коду.*

**БУЛО:**

private void pictureBox13\_Click(object sender, EventArgs e)

{

flowLayoutPanel1.Visible = true;

textBox1.Focus();

dt\_provider = new DataTable();

OleDbCommand selectProvider = new OleDbCommand();

selectProvider.Connection = myConnection;

selectProvider.CommandText = "SELECT \* FROM Постачальники";

selectProvider.CommandType = CommandType.Text;

adapter\_provider = new OleDbDataAdapter(selectProvider);

try

{

adapter\_provider.Fill(dt\_provider);

}

catch (OleDbException exc)

{

dt\_provider = null;

MessageBox.Show(exc.Message, "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

selectProvider.Connection.Close();

}

dt\_provider.PrimaryKey = new DataColumn[]

{

dt\_provider.Columns["Адреса"]

};

}

private void pictureBox14\_Click(object sender, EventArgs e)

{

flowLayoutPanel2.Visible = true;

textBox5.Focus();

OleDbCommand selectClient = new OleDbCommand();

selectClient.Connection = myConnection;

selectClient.CommandText = "SELECT \* FROM Замовники";

selectClient.CommandType = CommandType.Text;

adapter\_client = new OleDbDataAdapter(selectClient);

dt\_client = new DataTable();

try

{

adapter\_client.Fill(dt\_client);

}

catch (OleDbException exc)

{

dt\_client = null;

MessageBox.Show(exc.Message, "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

selectClient.Connection.Close();

}

dt\_client.PrimaryKey = new DataColumn[]

{

dt\_client.Columns["Адреса"]

};

}

**СТАЛО**:

public Tuple<DataTable, OleDbDataAdapter> Creation\_datatable(DataTable dt, OleDbDataAdapter adapter, OleDbConnection myConnection, string sqlselect)

{

dt = new DataTable();

OleDbCommand selectProvider = new OleDbCommand();

selectProvider.Connection = myConnection;

selectProvider.CommandText = sqlselect;

selectProvider.CommandType = CommandType.Text;

adapter = new OleDbDataAdapter(selectProvider);

try

{

adapter.Fill(dt);

}

catch (OleDbException exc)

{

dt = null;

MessageBox.Show(exc.Message, "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

selectProvider.Connection.Close();

}

dt.PrimaryKey = new DataColumn[]

{

dt.Columns["Адреса"]

};

return Tuple.Create(dt, adapter);

}

private void Form1\_Load(object sender, EventArgs e)

{

string sql\_select = "SELECT \* FROM Постачальники";

(dt\_provider, adapter\_provider) = Creation\_datatable(dt\_provider, adapter\_provider, myConnection, sql\_select);

string sql\_select2 = "SELECT \* FROM Замовники";

(dt\_client, adapter\_client) = Creation\_datatable(dt\_client, adapter\_client, myConnection, sql\_select2);

}

**Висновок**: в процесі виконання лабораторної роботи я навчилась виконувати реорганізацію програмного коду на підставі підбору шаблонів рефакторингу